PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

REC'D 13 APR 2000

WIPO   PCT

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.
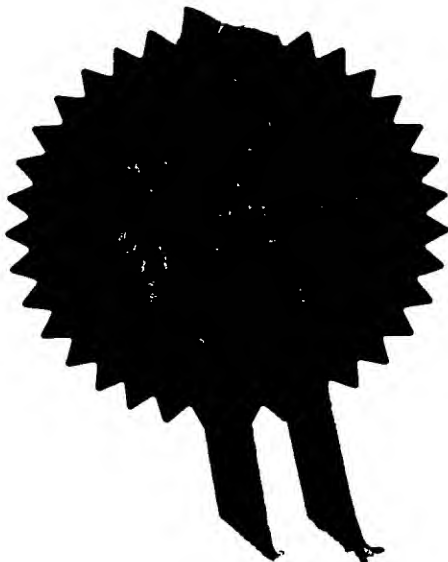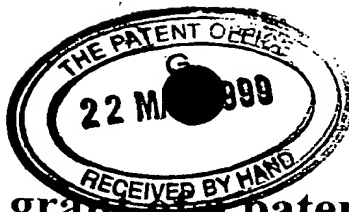
Signed

Dated   05 APR 2000

THIS PAGE BLANK (USPTO)

Request for grant of a patent

*(See the notes on the back of this form. You can also get
an explanatory leaflet from the Patent Office to help
you fill in this form.)*

The
**Patent
Office**

1/77

Fee: £0

1.	Your reference	AJR/40522

2.	Patent application number *(The Patent Office will fill in this part)*	**9906603.7**	**2 2 MAR 1999**

3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	VIDEOLOGIC LIMITED of Home Park Estate Kings Langley Hertfordshire WD4 8LZ UNITED KINGDOM
	Patents ADP number *(if you know it)*	438749002
	If the applicant is a corporate body, give the country/state of incorporation	UNITED KINGDOM

4.	Title of the invention	IMAGE COMPRESSION AND DECOMPRESSION

5.	Full name, address and postcode in the United Kingdom to which all correspondence relating to this form and translation should be sent	Reddie & Grose 16 Theobalds Road LONDON WC1X 8PL
	Patents ADP number *(if you know it)*	91001

		Country	Priority application *(If you know it)*	Date of filing *(day/month/year)*
6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number			

		Number of earlier application	Date of filing *(day/month/year)*
7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application		

8.	Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:* a) *any applicant named in part 3 is not an inventor, or* b) *there is an inventor who is not named as an applicant, or* c) *any named applicant is a corporate body.* *See note (d))*	YES

9. Enter number of sheets for any of the
    following items you are filing with this form.
    Do not count copies of the same document.

Continuation sheets of this form

Description	11
Claim *(s)*	3
Abstract	
Drawing *(s)*	2 ۱ 2 *16*

10. If you are also filing any of the following,
    state how many against each item.

Priority documents

Translations of priority documents

Statement of inventorship and right
to grant of a patent *(Patents Form 7/77)*

Request for preliminary examination
and search *(Patents Form 9/77)*   1

Request for substantive examination
(Patents Form 10/77)

Any other documents
(please specify)

11.   I/We request the grant of a patent on the basis of this application.

Signature   Date

                                       22 March 1999

12. Name and daytime telephone number of   A J ROBSON
    person to contact in the United Kingdom   0171-242 0901

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or such direction has been revoked.

Notes

a)   If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.

b)   Write your answers in capital letters using black ink or you may type them.

c)   If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.

d)   If you have answered 'Yes' Patents Form 7/77 will need to be filed.

e)   Once you have filled in the form you must remember to sign and date it.

f)   For details of the fee and ways to pay please contact the Patent Office.

IMAGE COMPRESSION AND DECOMPRESSION

This invention relates to image compression and decompression and is particularly applicable to compression and decompression techniques of types similar to the S3 compression technology licenced by Microsoft.

Image compression consists of making a new representation of the image data in digital form that takes up less storage space than the original. Many compression techniques are referred to as being lossy because they do not recover all the original data but instead make some approximation to this.

Often, the pixels that represent an image are stored in what is known as RGB format. This means they are built from three multi bit numbers which respectively represent the amount of red green and blue colours to be applied to a particular pixel. Another colour format is known as YUV. In this colour is encoded such that the brightness of the image is represented by the Y value and can be used to display black and white images whilst the colour is contained in the U and V values. These colour formats can be graphically represented using an orthognal coordinate system to define what is known as colour space.

In some cases translucency data is included with image data. This information is used when one image is shown against another image or a background and the translucency or alpha information is then used to combine one image with the other. This is particularly useful when blending the edge of one image into another. A special type of translucency known as punch-through can be used to switch on and off translucency whilst full variable translucency can be used for objects such as perspex which might have fractional or variable alpha values.

In most compression techniques blocks of pixels are operated on by the compression technique. In this patent

application we will usually be referring to a 4x4 pixel block although the invention may of course apply to any block size.

The S3 compression technology referred to above provides a technique of lossy image compression. It contains three different compression schemes known as DXT1, DXT2, and DXT3. The first of these is in principle an opaque method of compression in that its main role is to compress RGB pixels. It does cater for a particular type of translucency known as "punch through" which is used when translucency or alpha value is either on or off.

The other two compression schemes DXT2 and DXT3 are used only for translucency information where the alpha value is between 1 and 0.

DXT1 operates by first splitting the image into blocks of 4x4 pixels. Colours of each of the pixels in the block are analysed and the two points furthest apart in colour space are derived from this. Each of these end points is then represented by a 16 bit colour value in format RGB565 (meaning 5, 6, and 5 bits to each of the R, G and B channels respectively). An index of two bits per pixel is then used to assign a colour value along the line joining the two end points in colour space to each of the pixels in the 4x4 block. The possible values of the index for each pixel are 00, 01, 10, and 11. 00 represents one end point and 11 the other. The two other values represent intermediate points along the line which are derived by interpolation during decompression.

The choice of representative colours need not be the two furthest apart values. More complex schemes which iterate to reduce errors could be used, and weightings based on e.g. colour perception and other factors could be applied.

Index values are assigned to each pixel at the compressor by determining which of the four possible colour values corresponds most closely to the original colour value of that pixel. Thus, DXT1 provides a scheme

whereby a block of 4x4 pixels which each originally were represented by, for example, a 24 bit RGB colour value have their image data compressed to two 16 bit colour values and an index of two bits per pixel thereby comprising a total of 64 bits.

Punch through translucency is dealt with in DXT1 by having only one intermediate colour value in colour space and using the fourth index value to determine whether or not the translucency is on or off for a particular pixel. Thus, there is a reduction in the number of available colours for the 4x4 pixel block and the quality of the representation of colour in the block is thereby reduced when punch-through is selected.

DXT2 is used purely for translucency. It operates merely by restoring the alpha information at a lower precision. Therefore, it does not work particularly well on alpha layers that are stored at high precision.

DXT3 is another compression scheme for storing the alpha data. It does this in a similar manner to the colour method described above. For each 4x4 block two 8 bit alpha values are stored along with a 3 bit per pixel index to select between the two 8 bit alpha values and six interpolated values between these two extremes.

Problems with image quality can arise in schemes such as DXT1 described above when an image block contains an edge between two different colours. As the colour data for the block is generated by only two representative colours it is only possible to represent graduations between the two colours and not graduations of the two colours. Hence the scheme does not compress the image so well and in some cases this can result in the decompressed image having a blocky appearance. This is generally not acceptable.

Preferred embodiments of the present invention seek to reduce the blockiness of schemes such as DXT1 and to improve the visual quality of the image after compression and decompression by reducing visual artefacts as well as

by reducing more easily measured criteria such as the
least squares error.

　　　　Embodiments of the invention can also be applied to
the other DXT compressions schemes described above which
5　　are used for translucency information.

　　　　The invention is designed with more precision in the
appended claims to which reference should now be made.

　　　　A preferred embodiment of the invention will now be
described in detail by way of example with reference to
10　　the accompanying drawings in which:

　　　　Figure 1 is block diagram of a compression unit
embodying the invention; and

　　　　Figure 2 is a block diagram of a decompression unit
embodying the invention.

15　　　　　The embodiment described comprises an improvement to
the quality of the DXT1 image compression. The basic
scheme as described above compresses RGB colour data for a
4x4 pixel block into two RGB565 representative colours
totalling 32 bits and a 2 bit index per pixel indicating
20　　which interpolated colours to use. This makes another 32
bits giving 64 bits in all.

　　　　In Figure 1 which shows a block diagram of the
compression stage it can be seen that the original pixels
a enter the unit and are compressed according to the Dxt1
25　　scheme in the S3 compression unit 2 which outputs a
compressed approximation layer f for each block.

　　　　In this embodiment the unit 2 also outputs the
compression approximation layer at output b to be supplied
to a decompression unit 4. This decompresses the
30　　compressed approximation layer in the same manner as it
would be decompressed normally. The output of this c is
the data which would be used to display the original image
in decompressed form. This data and the original pixel
data a is fed to a subtraction unit 6. This is used to

derive a difference between the original pixels and the decompressed pixels.

Thus, for each pixel there will be a difference for each of the RGB values and this is provided at an output d. This difference value then passes to a transformation unit 8. This is optional but it will be used in a preferred embodiment. The purpose of this transformation unit is to determine the range of error values for each 4x4 block and to apply a scaling factor to them to ensure that they can subsequently be compressed and transmitted with the compressed image data in a form which will give reasonable accuracy on decompression. The scaling factor applied to the block is provided at an output g.

The other output of the transformation unit is the scale values themselves. These are passed to another DXT1 compression unit which compresses the difference pixel data in a manner similar to that of the approximation layer. Two representative difference colours are selected to represent the extremes in colour space of all the difference pixels in the block and are stored in RGB565 format. A 2 bit per pixel index is then used to indicate which out of these two end points and two intermediate points on the line joining these end points should be assigned to each pixel in the block.

Thus, a compressed error layer h is output by unit 10 comprising a further 64 bits in addition to the 64 bits of the compressed approximation layer. The compressed approximation layer, the compressed error layer, and the scaling factor for each block can then be stored or transmitted as required for subsequent decompression.

The system may be simplified by omitting the transformation unit and sending unscaled data.

The number of bits used for the error layer could be greater or fewer than that used for the compressed approximation layer. The bits used to represent the scaling factor are included in the approximation and error layers. The first bit comes from the ordering of

40522.uk

representatives in the approximation layer, e.g. ascending order denotes 1 and descending denotes 0. Similarly the second bit comes from the ordering of representatives in the error layer.

5   The image data is decompressed in the unit of Figure 2. This operates in the reverse manner to the compression unit.

The approximation layer f passes to a DXT1 decompression unit 12. That provides a decompressed

10   output k. At the same time, the error layer for the equivalent block passes to a DXT1 decompression unit 14 which provides decompressed errors i for each pixel in the block. The scaling factor g passes to the transformation unit 16 which performs the reverse scaling of the

15   transformation unit 8 on the scaled decompressed pixels to provide the unscaled decompressed errors j (this is omitted if unscaled data is sent). The decompressed pixel data k and decompressed error data j is then combined for each pixel in an addition unit 18 to give decompressed and

20   error adjusted pixels at output l.

The scaling factor stored can scale the error values by a fixed amount per block. This enables the method to dynamically change the range of errors stored in the different block h thereby enabling maximum resolution to

25   be given to each error. The system could be modified to provide a different scaling factor for each of the RG and B values.

In this particular implementation, two bits per block are used to specify which scaling is used out of a set of

30   eg. 16, 32, 64 and 128, which we refer to as the scaling set. The source of these bits is discussed below.

For example, if the original RGB values in the pixel are (76, 87, 129) and the approximation layer produces a pixel with value (62, 92, 116) after decompression, then

35   the difference pixel has a value of (6, 5, -13). As the maximum absolute difference for each colour component is less than 16, it is applicable to choose the range 16 from

the scaling set to represent all of the errors. The range
represented by the number 16 can represent any value
between -15 and 15, and hence the value 15, or in the
general case the value (range -1) is added to the
difference values to give a scaled difference pixel with
values (21, 20, 2) thus, each component is in the range 0-
31 or 0-(range -1). The number is then scaled so that it
occupies the highest possible range of, in this case, 8
bits. In this case, each component will be multiplied by
8 for scaling purposes. Thus, the scaled and translated
difference pixel is (168, 160, 16). With all 16 of the
transformed difference pixels in the block being found
they are then compressed using the DXT1 method.

Other scaling sets and methods for selecting scaling
functions can be used, e.g. maximums, average, or some
other measure.

The reverse process follows in the decompression
unit. If the decompressed difference pixel of the same
pixel as above is (175, 155, 20), then the scaled back
pixel is given by (21, 19, 3). This then gives the
decompressed difference pixel as (6, 4, -12). This can
then be subtracted from the decompressed values of the
original pixel, namely (62, 92, 116) to give (56, 88,
128). Which is much closer to the original pixel value of
(56, 87, 129) than the pixel value in the approximation
layer.

It is possible that some values may exceed the
largest possible difference value which here is 127. In
such a case the errors would be clamped to be within this
range.

The two bits that are used to choose the scaling
factor from the scaling set are chosen in the same way as
the S3 DXT1 compression scheme chooses between regular and
punch through blocks. This involves placing the two
representative values in either ascending or descending
order to indicate whether or not a scaling factor is
present, as explained above.

The compression unit shown in Figure 1 could be modified so that only one DXT1 compression unit is required. This would involve the use of a feedback mechanism to provide the compressed error layer from

5   compression unit 2 as well as the compressed approximation layer. Similarly, in decompression it is possible to use only one decompression unit by providing suitable multiplexing between the error approximation layers and a decompression unit and the transformation unit 16.

10   The compression scheme can in an alternative embodiment be used with punch through translucency. In this, one value of the index represents a punch through value. The fact that this value exists in the index is represented by the order of the two representative colours

15   or the two error values. If the first representative colour or error value stored is a larger value than the second then one of the index values represents punch through. If the representative colours or error values are in the opposite order then all the index values are

20   used to represent points on the line in colour space. When this is used then the reduction in quality is much less than in the basic DXT1 compression scheme because of the error correction process provided in the embodiment of the invention. In such an arrangement, the two bits used

25   for the scaling function are then provided by using RGB 555 instead of RGB 565 for the approximation error layer representatives. This frees up an additional 2 bits for use as the scaling function. Clearly, the punch through bit could be one of the bits released by RGB 555, as

30   desired in a particular application.

In another embodiment of the invention two layers are again used for transmission. The first of these consists of four pixels which are stored in colour format RGB 444 which takes up total storage of 48 bits. These pixels

35   each represent the average colour in each of the four 2x2 blocks in the 4x4 block. These average pixels are stored explicitly after truncation to RGB 444 format. The second

layer then consist of an error or difference layer which
is added to each of the average pixels for correction to a
closer approximation to each of the original pixel values.
A scaling is applied to each 2x2 sub block to bring all
5   the errors in each sub block into the same range in a
manner similar to that described above.

In another embodiment, an error layer is added to the
DXT2 or DXT3 schemes for compressing the alpha or
translucency value. The method by which this is done is
10   analogous to the method described above. In this, an
error layer is added to each DXT3 compression scheme. As
described above this scheme produces two 8 bit alpha
values for each sub block with an index of 3 bits to a
pixel to represent intermediate alpha values which are
15   generated from the representative values. This is
modified such that the approximation layer consists of two
8 bit alpha values followed by an index of 2 bits per
pixel instead of three. The error layer then consists of
two 8 bit difference representatives and an index of 2
20   bits per pixel to point to a total of 4 interpolated
difference values generated from the two representative
alpha values. The error per block can then be scaled
based on the average or maximum error per block and
truncated before compression to give better resolution in
25   blocks with small errors. As in the opaque case, 2 bits
resulting from ordering of the representatives can be used
to specify a scaling value from a scaling set of size 4.

The invention can be used on pixels which are stored
in other colour formats such as YUV. When this is done it
30   can be helpful to split the pixels up into different
groups as motivated by the sections of colour components.
Pixels stored in YUV format can be represented using three
schemes. Firstly, an approximation of the Y channel is
provided using two representatives in a manner similar to
35   the alpha only compression method above and a 2 bit index
per pixel. Then a second approximation layer for the U
and V channels is provided again using two UV

representatives and a 2 bit index per pixel. Finally, the
Y approximation layer is corrected with the layer
consisting of an error in Y. In this example the UV
layers not corrected an it is deemed less important
5   although this could be provided in another embodiment of
the invention.

        Finding the best two representatives or end points
for the pixel error and approximation layers need not
necessarily be done by choosing two furthest apart points
10   in colour space. Some form of training and iteration can
be used as follows.

        Firstly, the best two representatives or end points
for the pixels are found without considering the
interpolated points between them. This is done by taking
15   the two furthest apart pixels in colour space and then
performing a small number of generalised Lloyd algorithm
iterations to find a better pair of representatives.
These are the initial configuration or end point values.
Alternatively, we could simply start with the furthest of
20   pixels.

        Another way to find the end point codes is to do some
multivariate analysis on the pixels, and find the mean and
principal direction of the data by taking the eiganvalues
of the covariates matrix of the points. At this point, the
25   codes can be accepted and the next stage begins.

        Alternatively, a second stage of iteration can be
performed by doing a number of iterations where the best
fit two end point codes are improved upon. This is done
to satisfy a least squares condition with respect to the
30   nearest interpretted codes to the pixels at that given
iteration. Finally, the iteration could include some
simulated annealing to try and escape the local minima
instead of just accepting smaller errors.

        Other arrangements could be implemented such as
35   providing an approximation of YUV representatives
corrected only with a layer consisting of errors or
differences in Y.

This type of image compression and decompression is typically used in video graphics such as eg computer games for use in the home and supplied on some form of machine readable data carrier along with software relating to the

5   game.   It can be also be used in larger purpose built arcade type video games.   It could also be used in realtime video transmissions.

In the games type applications the compression of data is performed before the software is sold. Thus, the

10   compression unit such as the S3 unit would be modified in accordance with an embodiment of the invention either in hardware to provide a unit as shown in figure 1 above or, if all compression is provided in software this would be done by modifying the software used to compress images

15   using S3 technology.

40522.uk

CLAIMS

1.   A method of compressing a digital image data comprising the steps of:

compressing the image data using a predetermined compression technique;

decompressing the thus compressed image;

deriving difference values between the original image and the decompressed image;

compressing the thus derived difference values for use in subsequent correcting of the decompressed image; and

providing compressed image data and compressed difference values for decompression.

2.   A method according to claim 1 in which the difference values are compressed using the same compression method as the image.

3.   A method according to claim 1 and 2 in which the difference values have a scaling factor applied to them prior to compression and wherein the scaling factor is also provided for decompression.

4.   A method according to claim 1, 2, or 3 in which the image data comprises colour data.

5.   A method according to claim 1, 2, 3 or 4 in which the image data comprises translucency data.

6.   Apparatus for compressing digital image data comprising;

means for compressing the image data using a predetermined compressing technique;

means for decompressing the compressed image data;

means for deriving a difference value from the original image data and the decompressed image data;

means for compressing the thus derived difference values;   and

5   means for providing the compressed image data and compressed difference values for subsequent decompression.

7.   Apparatus according to claim 6 in which the means for compressing the difference values uses the same

10   compression technique as the means for compressing the image data.

8.   Apparatus according to claim 6 and 7 including means for scaling the difference values prior to compression and wherein the scaling factor is also

15   provided for decompression.

9.   A method for decompressing compressed digital image data comprising the steps of:

decompressing the compressed image data using a predetermined decompressing technique;

20   decompressing compressed difference values associated with the compressed image data; and

correcting the decompressed image data with the decompressed difference values.

10.   A method according to claim 9 in which the

25   compressed image data and difference values are both decompressed using the same decompression technique.

11.   A method according to claim 9 and 10 in which the difference values are scaled difference values and the method further includes the step of applying a reverse

30   scaling factor to the decompressed difference values.

40522.uk

12.   Apparatus for decompressing compressed digital
image data comprising:
            means for decompressing the compressed image
data according to a predetermined decompression technique;
5   means for decompressing compressed difference
values associated with the compressed image data; and
            means for correcting the decompressed image
data with the decompressed difference values.

13.   Apparatus according to claim 12 in which the
10   means for decompressing the image dat and the means for
decompressing the difference values both use the same
decompression technique.

14.   Apparatus according to claim 12 or 13 in which
the difference values are scaled difference values and
15   further comprising means for applying a reverse scaling
factor to the scaled difference values.

15.   A computer program product comprising image
data compressed according to the method of claim 1, 2, 3,
4, or 5.

20   16.   A machine readable data carrier comprising
image data compressed according to the method of claim 1,
2, 3, 4, or 5.

17.   A computer program product comprising a set of
instructions to configure a computer to compress digital
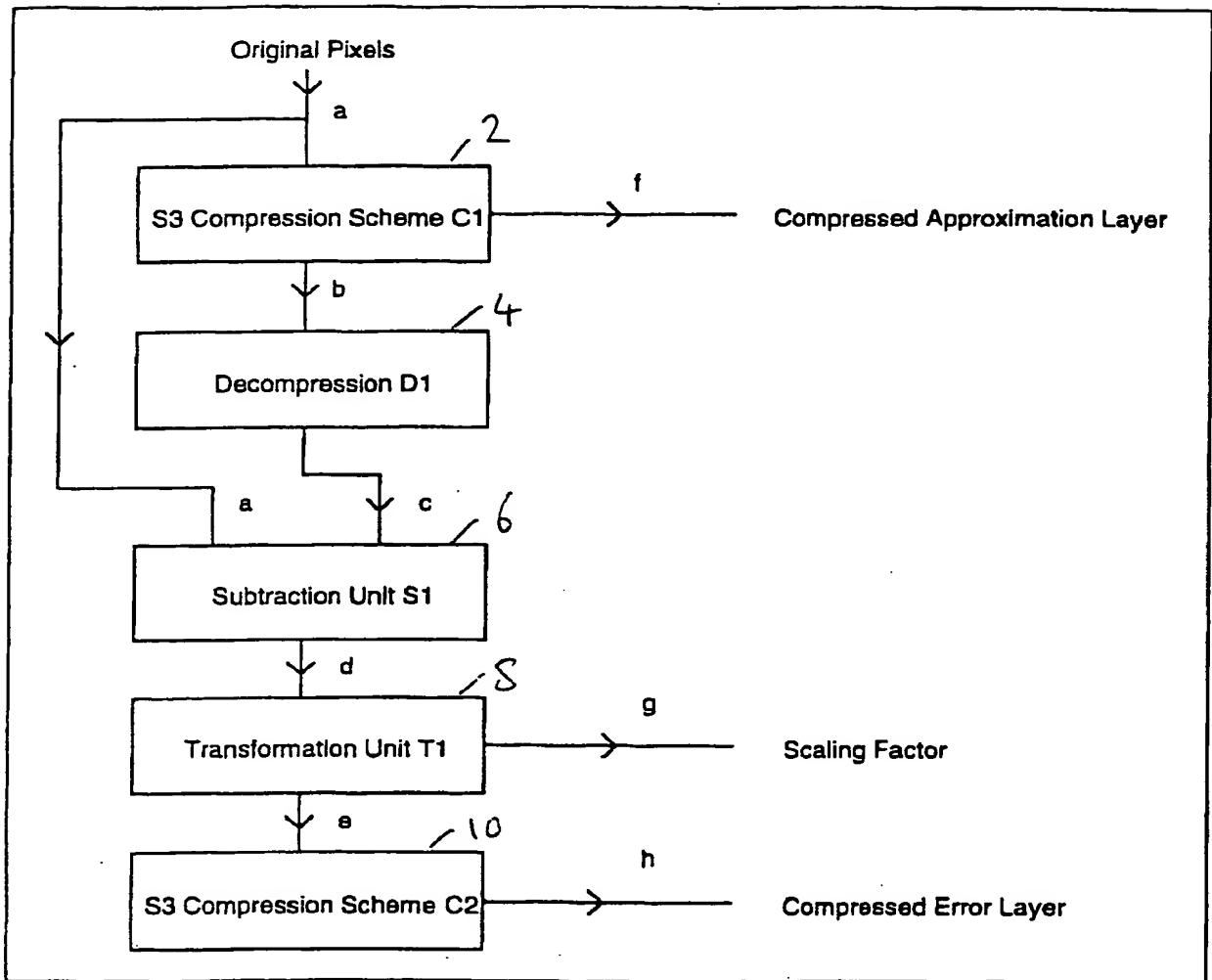25   image data according to the method of claim 1, 2, 3, 4, or
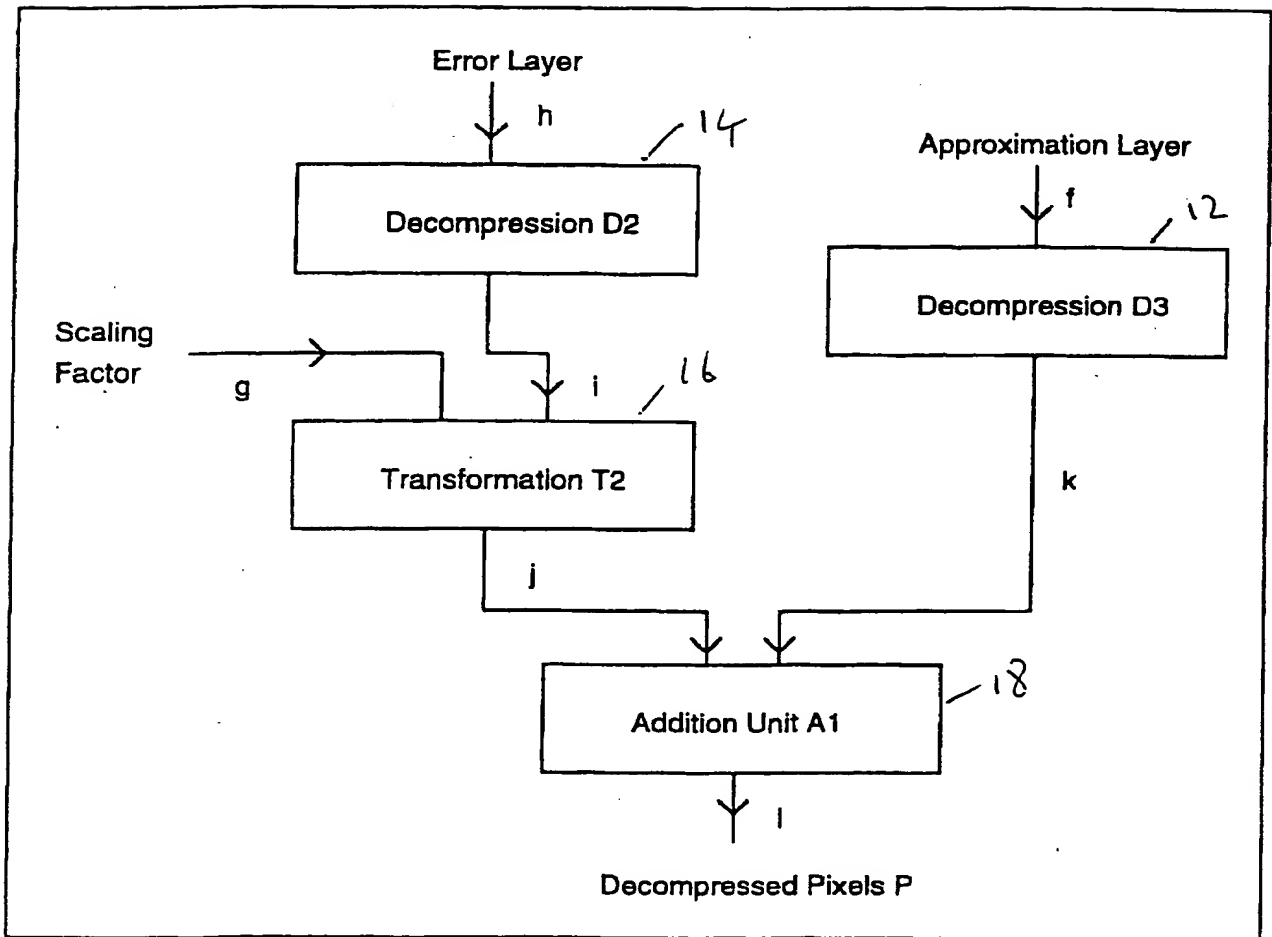5.

FIG. 1

FIG. 2

IMAGE COMPRESSION AND DECOMPRESSION

This invention relates to image compression and decompression and is particularly applicable to compression and decompression techniques of types similar

5   to the S3 compression technology licenced by Microsoft.

Image compression consists of making a new representation of the image data in digital form that takes up less storage space than the original. Many compression techniques are referred to as being lossy

10   because they do not recover all the original data but instead make some approximation to this.

Often, the pixels that represent an image are stored in what is known as RGB format. This means they are built from three multi bit numbers which respectively represent

15   the amount of red green and blue colours to be applied to a particular pixel. Another colour format is known as YUV. In this colour is encoded such that the brightness of the image is represented by the Y value and can be used to display black and white images whilst the colour is

20   contained in the U and V values. These colour formats can be graphically represented using an orthognal coordinate system to define what is known as colour space.

In some cases translucency data is included with image data. This information is used when one image is

25   shown against another image or a background and the translucency or alpha information is then used to combine one image with the other. This is particularly useful when blending the edge of one image into another. A special type of translucency known as punch-through can be

30   used to switch on and off translucency whilst full variable translucency can be used for objects such as perspex which might have fractional or variable alpha values.

In most compression techniques blocks of pixels are

35   operated on by the compression technique. In this patent

40522.uk

application we will usually be referring to a 4x4 pixel
block although the invention may of course apply to any
block size.

The S3 compression technology referred to above
provides a technique of lossy image compression. It
contains three different compression schemes known as
DXT1, DXT2, and DXT3. The first of these is in principle
an opaque method of compression in that its main role is
to compress RGB pixels. It does cater for a particular
type of translucency known as "punch through" which is
used when translucency or alpha value is either on or off.

The other two compression schemes DXT2 and DXT3 are
used only for translucency information where the alpha
value is between 1 and 0.

DXT1 operates by first splitting the image into
blocks of 4x4 pixels. Colours of each of the pixels in
the block are analysed and the two points furthest apart
in colour space are derived from this. Each of these end
points is then represented by a 16 bit colour value in
format RGB565 (meaning 5, 6, and 5 bits to each of the R,
G and B channels respectively). An index of two bits per
pixel is then used to assign a colour value along the line
joining the two end points in colour space to each of the
pixels in the 4x4 block. The possible values of the index
for each pixel are 00, 01, 10, and 11. 00 represents one
end point and 11 the other. The two other values
represent intermediate points along the line which are
derived by interpolation during decompression.

The choice of representative colours need not be the
two furthest apart values. More complex schemes which
iterate to reduce errors could be used, and weightings
based on e.g. colour perception and other factors could be
applied.

Index values are assigned to each pixel at the
compressor by determining which of the four possible
colour values corresponds most closely to the original
colour value of that pixel. Thus, DXT1 provides a scheme

whereby a block of 4x4 pixels which each originally were represented by, for example, a 24 bit RGB colour value have their image data compressed to two 16 bit colour values and an index of two bits per pixel thereby comprising a total of 64 bits.

Punch through translucency is dealt with in DXT1 by having only one intermediate colour value in colour space and using the fourth index value to determine whether or not the translucency is on or off for a particular pixel. Thus, there is a reduction in the number of available colours for the 4x4 pixel block and the quality of the representation of colour in the block is thereby reduced when punch-through is selected.

DXT2 is used purely for translucency. It operates merely by restoring the alpha information at a lower precision. Therefore, it does not work particularly well on alpha layers that are stored at high precision.

DXT3 is another compression scheme for storing the alpha data. It does this in a similar manner to the colour method described above. For each 4x4 block two 8 bit alpha values are stored along with a 3 bit per pixel index to select between the two 8 bit alpha values and six interpolated values between these two extremes.

Problems with image quality can arise in schemes such as DXT1 described above when an image block contains an edge between two different colours. As the colour data for the block is generated by only two representative colours it is only possible to represent graduations between the two colours and not graduations of the two colours. Hence the scheme does not compress the image so well and in some cases this can result in the decompressed image having a blocky appearance. This is generally not acceptable.

Preferred embodiments of the present invention seek to reduce the blockiness of schemes such as DXT1 and to improve the visual quality of the image after compression and decompression by reducing visual artefacts as well as

by reducing more easily measured criteria such as the
least squares error.

Embodiments of the invention can also be applied to
the other DXT compressions schemes described above which
are used for translucency information.

The invention is designed with more precision in the
appended claims to which reference should now be made.

A preferred embodiment of the invention will now be
described in detail by way of example with reference to
the accompanying drawings in which:

Figure 1 is block diagram of a compression unit
embodying the invention; and

Figure 2 is a block diagram of a decompression unit
embodying the invention.

The embodiment described comprises an improvement to
the quality of the DXT1 image compression. The basic
scheme as described above compresses RGB colour data for a
4x4 pixel block into two RGB565 representative colours
totalling 32 bits and a 2 bit index per pixel indicating
which interpolated colours to use. This makes another 32
bits giving 64 bits in all.

In Figure 1 which shows a block diagram of the
compression stage it can be seen that the original pixels
a enter the unit and are compressed according to the Dxt1
scheme in the S3 compression unit 2 which outputs a
compressed approximation layer f for each block.

In this embodiment the unit 2 also outputs the
compression approximation layer at output b to be supplied
to a decompression unit 4. This decompresses the
compressed approximation layer in the same manner as it
would be decompressed normally. The output of this c is
the data which would be used to display the original image
in decompressed form. This data and the original pixel
data a is fed to a subtraction unit 6. This is used to

derive a difference between the original pixels and the decompressed pixels.

Thus, for each pixel there will be a difference for each of the RGB values and this is provided at an output
5   d.   This difference value then passes to a transformation unit 8.   This is optional but it will be used in a preferred embodiment.   The purpose of this transformation unit is to determine the range of error values for each 4x4 block and to apply a scaling factor to them to ensure
10   that they can subsequently be compressed and transmitted with the compressed image data in a form which will give reasonable accuracy on decompression.   The scaling factor applied to the block is provided at an output g.

The other output of the transformation unit is the
15   scale values themselves.   These are passed to another DXT1 compression unit which compresses the difference pixel data in a manner similar to that of the approximation layer.   Two representative difference colours are selected to represent the extremes in colour space of all the
20   difference pixels in the block and are stored in RGB565 format.   A 2 bit per pixel index is then used to indicate which out of these two end points and two intermediate points on the line joining these end points should be assigned to each pixel in the block.

25   Thus, a compressed error layer h is output by unit 10 comprising a further 64 bits in addition to the 64 bits of the compressed approximation layer.   The compressed approximation layer, the compressed error layer, and the scaling factor for each block can then be stored or
30   transmitted as required for subsequent decompression.

The system may be simplified by omitting the transformation unit and sending unscaled data.

The number of bits used for the error layer could be greater or fewer than that used for the compressed
35   approximation layer.   The bits used to represent the scaling factor are included in the approximation and error layers.   The first bit comes from the ordering of

representatives in the approximation layer, e.g. ascending
order denotes 1 and descending denotes 0. Similarly the
second bit comes from the ordering of representatives in
the error layer.

5   The image data is decompressed in the unit of Figure
2. This operates in the reverse manner to the compression
unit.

The approximation layer f passes to a DXT1
decompression unit 12. That provides a decompressed

10   output k. At the same time, the error layer for the
equivalent block passes to a DXT1 decompression unit 14
which provides decompressed errors i for each pixel in the
block. The scaling factor g passes to the transformation
unit 16 which performs the reverse scaling of the

15   transformation unit 8 on the scaled decompressed pixels to
provide the unscaled decompressed errors j (this is
omitted if unscaled data is sent). The decompressed pixel
data k and decompressed error data j is then combined for
each pixel in an addition unit 18 to give decompressed and

20   error adjusted pixels at output l.

The scaling factor stored can scale the error values
by a fixed amount per block. This enables the method to
dynamically change the range of errors stored in the
different block h thereby enabling maximum resolution to

25   be given to each error. The system could be modified to
provide a different scaling factor for each of the RG and
B values.

In this particular implementation, two bits per block
are used to specify which scaling is used out of a set of

30   eg. 16, 32, 64 and 128, which we refer to as the scaling
set. The source of these bits is discussed below.

For example, if the original RGB values in the pixel
are (76, 87, 129) and the approximation layer produces a
pixel with value (62, 92, 116) after decompression, then

35   the difference pixel has a value of (6, 5, -13). As the
maximum absolute difference for each colour component is
less than 16, it is applicable to choose the range 16 from

the scaling set to represent all of the errors. The range represented by the number 16 can represent any value between -15 and 15, and hence the value 15, or in the general case the value (range -1) is added to the difference values to give a scaled difference pixel with values (21, 20, 2) thus, each component is in the range 0-31 or 0-(range -1). The number is then scaled so that it occupies the highest possible range of, in this case, 8 bits. In this case, each component will be multiplied by 8 for scaling purposes. Thus, the scaled and translated difference pixel is (168, 160, 16). With all 16 of the transformed difference pixels in the block being found they are then compressed using the DXT1 method.

Other scaling sets and methods for selecting scaling functions can be used, e.g. maximums, average, or some other measure.

The reverse process follows in the decompression unit. If the decompressed difference pixel of the same pixel as above is (175, 155, 20), then the scaled back pixel is given by (21, 19, 3). This then gives the decompressed difference pixel as (6, 4, -12). This can then be subtracted from the decompressed values of the original pixel, namely (62, 92, 116) to give (56, 88, 128). Which is much closer to the original pixel value of (56, 87, 129) than the pixel value in the approximation layer.

It is possible that some values may exceed the largest possible difference value which here is 127. In such a case the errors would be clamped to be within this range.

The two bits that are used to choose the scaling factor from the scaling set are chosen in the same way as the S3 DXT1 compression scheme chooses between regular and punch through blocks. This involves placing the two representative values in either ascending or descending order to indicate whether or not a scaling factor is present, as explained above.

The compression unit shown in Figure 1 could be
modified so that only one DXT1 compression unit is
required.   This would involve the use of a feedback
mechanism to provide the compressed error layer from

5   compression unit 2 as well as the compressed approximation
layer.   Similarly, in decompression it is possible to use
only one decompression unit by providing suitable
multiplexing between the error approximation layers and a
decompression unit and the transformation unit 16.

10   The compression scheme can in an alternative
embodiment be used with punch through translucency.   In
this, one value of the index represents a punch through
value.   The fact that this value exists in the index is
represented by the order of the two representative colours

15   or the two error values.   If the first representative
colour or error value stored is a larger value than the
second then one of the index values represents punch
through.   If the representative colours or error values
are in the opposite order then all the index values are

20   used to represent points on the line in colour space.
When this is used then the reduction in quality is much
less than in the basic DXT1 compression scheme because of
the error correction process provided in the embodiment of
the invention.   In such an arrangement, the two bits used

25   for the scaling function are then provided by using RGB
555 instead of RGB 565 for the approximation error layer
representatives.   This frees up an additional 2 bits for
use as the scaling function.   Clearly, the punch through
bit could be one of the bits released by RGB 555, as

30   desired in a particular application.

In another embodiment of the invention two layers are
again used for transmission.   The first of these consists
of four pixels which are stored in colour format RGB 444
which takes up total storage of 48 bits.   These pixels

35   each represent the average colour in each of the four 2x2
blocks in the 4x4 block.   These average pixels are stored
explicitly after truncation to RGB 444 format.   The second

layer then consist of an error or difference layer which
is added to each of the average pixels for correction to a
closer approximation to each of the original pixel values.
A scaling is applied to each 2x2 sub block to bring all
5   the errors in each sub block into the same range in a
manner similar to that described above.

In another embodiment, an error layer is added to the
DXT2 or DXT3 schemes for compressing the alpha or
translucency value. The method by which this is done is
10   analogous to the method described above. In this, an
error layer is added to each DXT3 compression scheme. As
described above this scheme produces two 8 bit alpha
values for each sub block with an index of 3 bits to a
pixel to represent intermediate alpha values which are
15   generated from the representative values. This is
modified such that the approximation layer consists of two
8 bit alpha values followed by an index of 2 bits per
pixel instead of three. The error layer then consists of
two 8 bit difference representatives and an index of 2
20   bits per pixel to point to a total of 4 interpolated
difference values generated from the two representative
alpha values. The error per block can then be scaled
based on the average or maximum error per block and
truncated before compression to give better resolution in
25   blocks with small errors. As in the opaque case, 2 bits
resulting from ordering of the representatives can be used
to specify a scaling value from a scaling set of size 4.

The invention can be used on pixels which are stored
in other colour formats such as YUV. When this is done it
30   can be helpful to split the pixels up into different
groups as motivated by the sections of colour components.
Pixels stored in YUV format can be represented using three
schemes. Firstly, an approximation of the Y channel is
provided using two representatives in a manner similar to
35   the alpha only compression method above and a 2 bit index
per pixel. Then a second approximation layer for the U
and V channels is provided again using two UV

representatives and a 2 bit index per pixel. Finally, the
Y approximation layer is corrected with the layer
consisting of an error in Y. In this example the UV
layers not corrected an it is deemed less important

5   although this could be provided in another embodiment of
the invention.

Finding the best two representatives or end points
for the pixel error and approximation layers need not
necessarily be done by choosing two furthest apart points

10   in colour space. Some form of training and iteration can
be used as follows.

Firstly, the best two representatives or end points
for the pixels are found without considering the
interpolated points between them. This is done by taking

15   the two furthest apart pixels in colour space and then
performing a small number of generalised Lloyd algorithm
iterations to find a better pair of representatives.
These are the initial configuration or end point values.
Alternatively, we could simply start with the furthest of

20   pixels.

Another way to find the end point codes is to do some
multivariate analysis on the pixels, and find the mean and
principal direction of the data by taking the eiganvalues
of the covariates matrix of the points. At this point, the

25   codes can be accepted and the next stage begins.

Alternatively, a second stage of iteration can be
performed by doing a number of iterations where the best
fit two end point codes are improved upon. This is done
to satisfy a least squares condition with respect to the

30   nearest interpretted codes to the pixels at that given
iteration. Finally, the iteration could include some
simulated annealing to try and escape the local minima
instead of just accepting smaller errors.

Other arrangements could be implemented such as

35   providing an approximation of YUV representatives
corrected only with a layer consisting of errors or
differences in Y.

This type of image compression and decompression is typically used in video graphics such as eg computer games for use in the home and supplied on some form of machine readable data carrier along with software relating to the

5   game. It can be also be used in larger purpose built arcade type video games. It could also be used in realtime video transmissions.

In the games type applications the compression of data is performed before the software is sold. Thus, the

10   compression unit such as the S3 unit would be modified in accordance with an embodiment of the invention either in hardware to provide a unit as shown in figure 1 above or, if all compression is provided in software this would be done by modifying the software used to compress images

15   using S3 technology.

CLAIMS

1. A method of compressing a digital image data comprising the steps of:

compressing the image data using a predetermined compression technique;

decompressing the thus compressed image;

deriving difference values between the original image and the decompressed image;

compressing the thus derived difference values for use in subsequent correcting of the decompressed image; and

providing compressed image data and compressed difference values for decompression.

2. A method according to claim 1 in which the difference values are compressed using the same compression method as the image.

3. A method according to claim 1 and 2 in which the difference values have a scaling factor applied to them prior to compression and wherein the scaling factor is also provided for decompression.

4. A method according to claim 1, 2, or 3 in which the image data comprises colour data.

5. A method according to claim 1, 2, 3 or 4 in which the image data comprises translucency data.

6. Apparatus for compressing digital image data comprising;

means for compressing the image data using a predetermined compressing technique;

means for decompressing the compressed image data;

means for deriving a difference value from the
original image data and the decompressed image data;

means for compressing the thus derived
difference values; and

5   means for providing the compressed image data
and compressed difference values for subsequent
decompression.

7.   Apparatus according to claim 6 in which the
means for compressing the difference values uses the same
10   compression technique as the means for compressing the
image data.

8.   Apparatus according to claim 6 and 7 including
means for scaling the difference values prior to
compression and wherein the scaling factor is also
15   provided for decompression.

9.   A method for decompressing compressed digital
image data comprising the steps of:

decompressing the compressed image data using a
predetermined decompressing technique;

20   decompressing compressed difference values
associated with the compressed image data; and

correcting the decompressed image data with the
decompressed difference values.

10.   A method according to claim 9 in which the
25   compressed image data and difference values are both
decompressed using the same decompression technique.

11.   A method according to claim 9 and 10 in which
the difference values are scaled difference values and the
method further includes the step of applying a reverse
30   scaling factor to the decompressed difference values.

12.   Apparatus for decompressing compressed digital
image data comprising:
           means for decompressing the compressed image
data according to a predetermined decompression technique;
5   means for decompressing compressed difference
values associated with the compressed image data; and
           means for correcting the decompressed image
data with the decompressed difference values.

13.   Apparatus according to claim 12 in which the
10 means for decompressing the image dat and the means for
decompressing the difference values both use the same
decompression technique.

14.   Apparatus according to claim 12 or 13 in which
the difference values are scaled difference values and
15 further comprising means for applying a reverse scaling
factor to the scaled difference values.

15.   A computer program product comprising image
data compressed according to the method of claim 1, 2, 3,
4, or 5.

20   16.   A machine readable data carrier comprising
image data compressed according to the method of claim 1,
2, 3, 4, or 5.

17.   A computer program product comprising a set of
instructions to configure a computer to compress digital
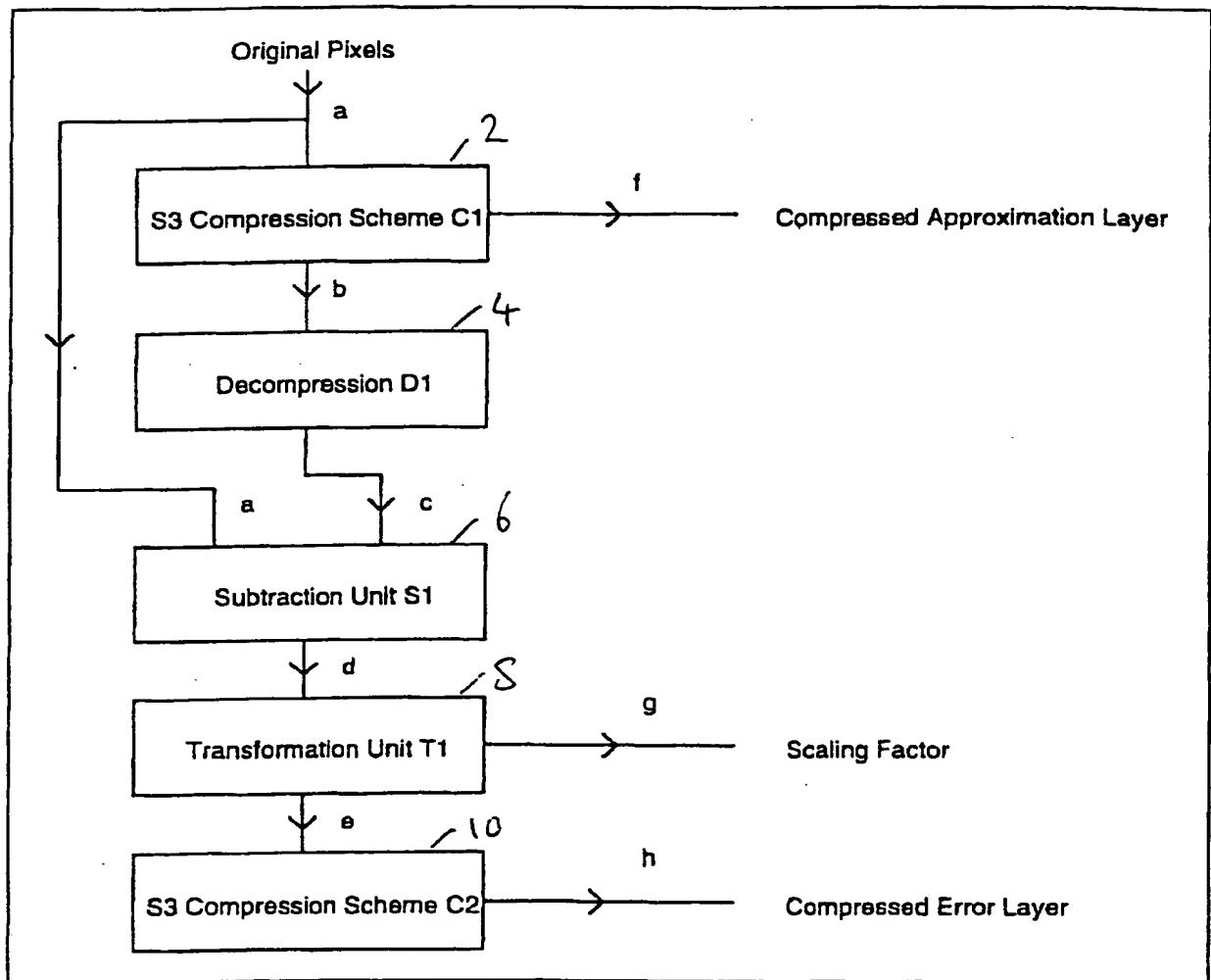25 image data according to the method of claim 1, 2, 3, 4, or
5.

Original Pixels

↓ a

S3 Compression Scheme C1 — 2 — f → Compressed Approximation Layer

↓ b

Decompression D1 — 4

a ↓ c — 6

Subtraction Unit S1

↓ d — 8

Transformation Unit T1 — g → Scaling Factor

↓ e — 10

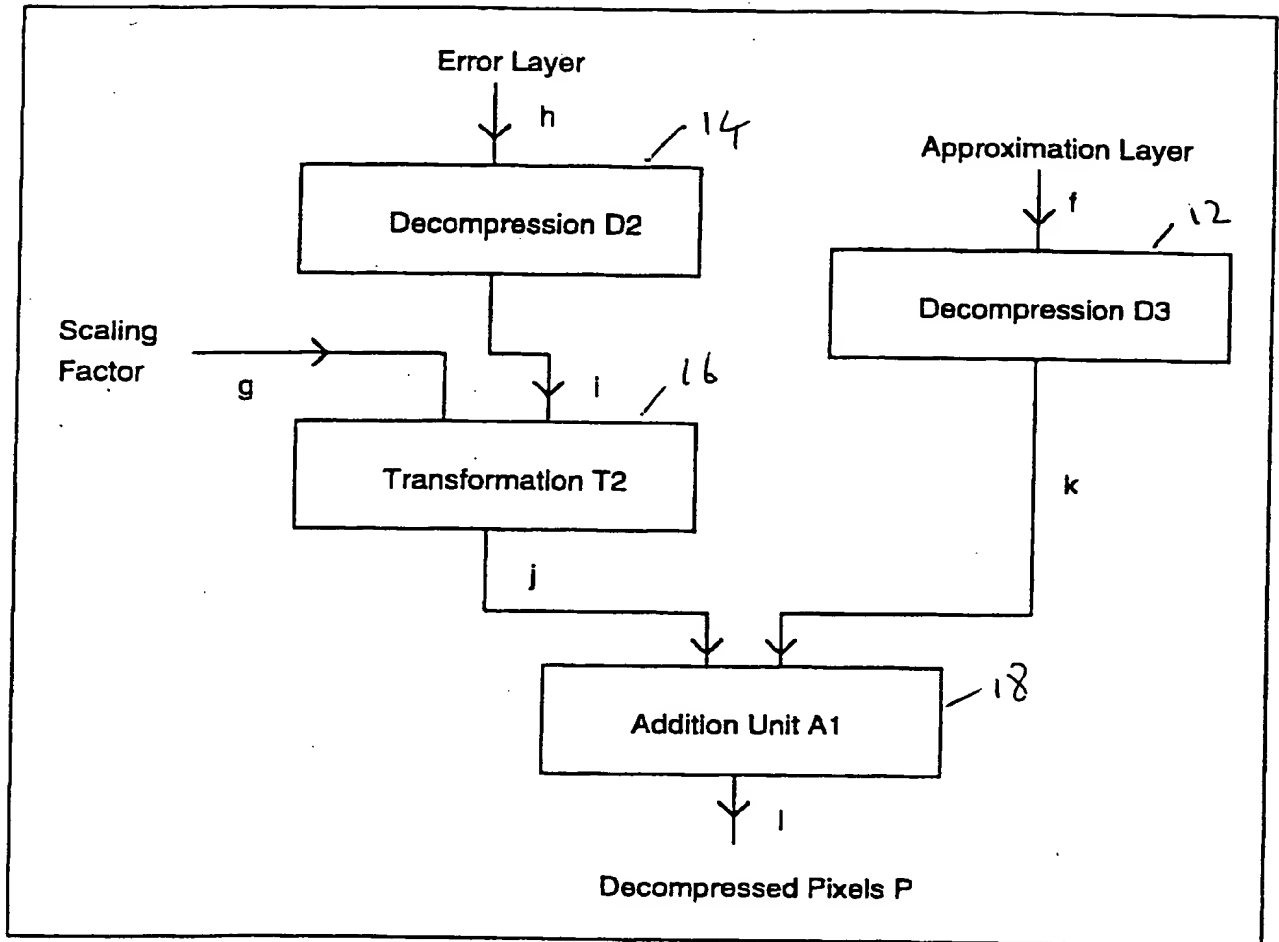S3 Compression Scheme C2 — h → Compressed Error Layer

FIG. 1

THIS PAGE BLANK (USPTO)

FIG. 2

22-3-00
Reddie & Grose
PCT/GB00/01081

THIS PAGE BLANK (USPTO)